# Task-agnostic object recognition for mobile robots through few-shot image matching

*electronics*

MDPI

*Article*

# Task-Agnostic Object Recognition for Mobile Robots through Few-Shot Image Matching

**Agnese Chiatti [1,*], Gianluca Bardaro [1], Emanuele Bastianelli [2], Ilaria Tiddi [3], Prasenjit Mitra [4] and Enrico Motta [1]**

1 Knowledge Media Institute, The Open University, Milton Keynes MK7 6AA, UK; gianluca.bardaro@open.ac.uk (G.B.); enrico.motta@open.ac.uk (E.M.)
2 The Interaction Lab, Heriot-Watt University, Edinburgh EH14 4AS, UK; e.bastianelli@hw.ac.uk
3 Faculty of Computer Science, Vrije Universitet Amsterdam, 1081 HV Amsterdam, The Netherlands; i.tiddi@vu.nl
4 Information Sciences and Technology, The Pennsylvania State University, University Park, PA 16801, USA; pmitra@ist.psu.edu
* Correspondence: agnese.chiatti@open.ac.uk; Tel.: +44-(0)1908-655115

Received: 29 November 2019; Accepted: 19 February 2020; Published: 25 February 2020


**Abstract:** To assist humans with their daily tasks, mobile robots are expected to navigate complex and dynamic environments, presenting unpredictable combinations of known and unknown objects. Most state-of-the-art object recognition methods are unsuitable for this scenario because they require that: (i) all target object classes are known beforehand, and (ii) a vast number of training examples is provided for each class. This evidence calls for novel methods to handle unknown object classes, for which fewer images are initially available (few-shot recognition). One way of tackling the problem is learning how to match novel objects to their most similar supporting example. Here, we compare different (shallow and deep) approaches to few-shot image matching on a novel data set, consisting of 2D views of common object types drawn from a combination of ShapeNet and Google. First, we assess if the similarity of objects learned from a combination of ShapeNet and Google can scale up to new object classes, i.e., categories unseen at training time. Furthermore, we show how normalising the learned embeddings can impact the generalisation abilities of the tested methods, in the context of two novel configurations: (i) where the weights of a Convolutional two-branch Network are imprinted and (ii) where the embeddings of a Convolutional Siamese Network are L2-normalised.

**Keywords:** few-shot object recognition; image matching; robotics

## 1. Introduction

As the fields of Artificial Intelligence (AI) and Robotics mature and evolve, an increasing number of hardware and software solutions have become available, reducing the costs and technical barriers of developing novel robotic platforms. Following these advancements, the members of the European Public-Private Partnership in Robotics (SPARC) have defined a set of critical objectives for the 2014–2020 horizon [1], which all relate to the integration of autonomous agents in the different urban sectors (e.g., Healthcare, Transport, Manufacturing, Commercial). This strategy also aligns with the vision of a Smart City, where robots carry out tasks in urban environments, while at the same time also acting as moving sensors, thus aiding the real-time collection of data about the city operations [2]. Differently from static sensors, robots can intervene in their surroundings (e.g., by navigating to specific locations, by grasping and manipulating objects, or by dialoguing with users). This characteristic makes them suitable to assist humans on tasks as diverse as health and safety monitoring [3], pre-emptive elderly care [4], and door-to-door garbage collection [5], just to name a few. To accomplish these tasks, the new

generation of robots is expected to make sense of environments which are always changing and evolving (e.g., cities or, at a more local scale, offices, warehouses, households and shops). Equipping robots with effective methods to recognise the different objects they encounter while navigating the environment is thus a crucial prerequisite to sense-making.

Let us consider the case of HanS, the Health and Safety autonomous inspector under development at the Knowledge Media Institute (KMi) [3]. HanS was conceived for autonomously identifying potentially hazardous situations, e.g., the fire hazard caused by a pile of papers sitting next to a portable heater. To identify the threat, HanS would first need to recognise the objects *paper* and *electric heater* autonomously. Furthermore, it will need to recognise that the two objects are next to each other. Based on that, HanS could then query external knowledge bases, such as ConceptNet [6] or DBpedia [7], which provide useful common-sense facts regarding the two objects, like paper is flammable and electric heater is a type of electrical device. With this knowledge, HanS would then be one step closer to verify that the health and safety rule stating that flammable items must be kept away from ignition sources has been violated.

This example explains why effective object recognition mechanisms are required to interpret the current status of the environment, i.e., to form the robot's worldview. Moreover, the top-performing object recognition methods available (e.g., [8–11]) (i) assume that all target object classes are known beforehand and (ii) require a vast number of annotated training examples to recognise each class. As a result, these methods do not align with the scenario of a learner who refines its worldview incrementally, based on the objects it already knows and with access to only a few instances of the newly-encountered objects (i.e., few-shot learning). In fact, HanS would need to learn how to spot papers and electric heaters from different viewpoints, under different conditions of lighting and clutter. It would also need to recognise new object types (e.g., people and chairs in addition to fire extinguishers and emergency exits) in order to manage different tasks (e.g., checking room occupancy as well as detecting fire hazards).

One way of tackling the few-shot recognition problem is learning how to match newly-incoming objects to their most similar support example [12,13]. These approaches are based on Convolutional Neural Networks (CNN) and typically reuse the knowledge acquired on large-scale benchmark datasets, such as ImageNet [14], to compensate for the lack of training examples. Inspired by the way cognitive imprinting works for humans, Qi et al. [15] have shown that the very first image available of a new object, if opportunely scaled and L2-normalised, can be used as enhanced weights in a multinomial classification layer, i.e., comprising of three or more classes. However, the weight imprinting strategy has yet to be integrated into a multi-branch CNN. Therefore, we investigate whether state-of-the-art methods for few-shot recognition that use multi-branch CNNs [13] can be further improved by introducing weight imprinting [15]. In the case of architectures including binary classifiers [12], we will instead study the impact of L2-normalising of the image embeddings.

Furthermore, the state-of-the-art few-shot image matching methods discussed in [13] have only been evaluated on specific tasks (e.g., stowing, grasping) and product domains (e.g., Amazon products). However, the recent availability of large-scale repositories of object models, such as ShapeNet [16], has opened up opportunities to evaluate how these methods can scale up to different object types, which represent object models agnostically to the specific product domain or end task. The fact that models in ShapeNet are annotated with respect to the WordNet taxonomy [17] also facilitates the integration of other knowledge bases in this process, which are linked to the same taxonomy, such as ConceptNet [6], DBpedia [7], or Visual Genome [18].

Based on these premises, we focus on the following contributions.

- We test state-of-the-art methods for Deep few-shot image matching on a novel task-agnostic data set, including 2D views of common object types drawn from a combination of ShapeNet [16] and Google Images.

- We compare these against other shallow methods explored as part of our prior work [19], which are based on colour and shape similarity, to assess whether meaningful features can be transferred after learning on a different image domain, i.e., ImageNet, without any prior feature engineering.
- We also evaluate the performance effects of (i) imprinting the weights [15] of a two-branch CNN [13], i.e., thus extending the framework in [13] with a novel weight configuration; (ii) applying L2 normalisation to the embeddings of a Convolutional Siamese Network [12], to include the case of binary classification in our assessment.

All the described data and code are available at our Github repository (https://github.com/kmi-robots/semantic-map-object-recognition/).

## 2. Background and Related Work

The development of service robots [3–5], has become a very active area of research, concurrently with the recent evolution of urban spaces to embrace the vision of Smart Cities, integrating different sensors, actuators and knowledge bases for the real-time management of the city operations [1,2]. This scenario calls for effective methods that can allow robots to make sense of fast-changing environments, presenting unpredictable combinations of known and novel objects.

The problem of real-time object recognition has reached satisfactory solutions [8–11] only in experimental scenarios where a very large amounts of human-annotated data is available and all object classes are assumed to be predetermined, also known as the closed world assumption [20].

Naturally, problems such as the paucity of training data or the adaptability to new learning environments are pervasive across all sub-fields of Artificial Intelligence (AI), and not only specific to the object recognition area. All efforts ever devoted towards tackling these problems are broadly ascribed to the Lifelong (Continual, or Never-ending) Machine Learning framework [20,21]. Lifelong Machine Learning (LML) is the term used to define the AI discipline concerned with the autonomous systems' capability to (i) improve at learning to perform a single task over time (Incremental Machine Learning), as well as (ii) reuse the acquired knowledge across different tasks [20]. The challenge of learning incrementally also implies to effectively manage the initial stages of learning, i.e., when a minimal number of training examples are likely to be available. Another objective is thus "learning to learn" (Meta-learning [20,22]), by focusing on how a new task can be learned effectively from a minimal number of training examples and, ideally, by producing task-agnostic models, i.e., models which can generalise to many different learning problems.

Most recently, the LML field has seen the rise of several methods based on Artificial Neural Networks (NN), given their promising performance on static tasks [21]. However, NNs are particularly prone to the issue of catastrophically forgetting previously-learned concepts, as soon as new concepts are introduced [23]. A typical, static setup usually entails: (i) shuffling the training examples to resemble iid conditions, (ii) randomly initialise the model parameters, and then (iii) gradually regularise them through successive full passes over the entire data set (epochs). This setup does not scale up to the case of learning in dynamic environments, where models need to be updated incrementally to reflect any changes in the environment, as well as the availability of new data points. These considerations fall under the so-called stability-plasticity dilemma, ascribed initially to Biological Neural systems [24] but also affecting Artificial NN design. In essence, a desirable architecture should offer a trade-off between the ability to encode new knowledge and the ability to retain prior knowledge.

In the attempt to achieve this trade-off, many strategies have been proposed [21,25–31], which can be grouped as replay-based, dynamic architectures and prior-focused [26]. Replay-based methods, as the name suggests, rely on jointly feeding the model with: (i) raw or compressed representations of previously-stored examples (i.e., memory replay), and (ii) newly-retrieved examples [25–27]. Within the second group of methods, instead, the building blocks of the Neural architecture are automatically re-organised at each update [21,28,29]. Lastly, prior-based methods use the model parameter distribution learned on certain tasks and data sets as prior for new learning tasks [30,31].

A common prior-based practice is to first initialise the model parameters with optimal values obtained on larger data collections and to then fine-tune the model [13,32].

In the specific context of NN-based object recognition, these broader issues translate into two main challenges: (i) building object representations which can be reused across different tasks and environmental conditions, (ii) even from the few training examples initially available (i.e., few shot object recognition).

With respect to the first challenge, the object vectorised representations produced by a Neural Network (i.e., embeddings) can be optimised either based on their classification among a set of pre-defined classes or by learning a feature space where similar objects are mapped closer to one another than dissimilar objects (a task often referred to as metric learning or image matching). One advantage of the latter training routines is that even novel objects, i.e., unseen at training time, can be classified at inference time, based on their nearest neighbours.

Nearest neighbour-based matching can also be related to cognitive studies on prototype-based object categorisation [33–35]. According to prototype theory, the cognitive process of categorisation (i.e., the human process of learning how to group objects into categories by visual stimuli) is achieved by converging towards a prototypical representation of all instances belonging to the same class, through abstraction. This abstract representation appears to be correlated with the measures of central tendency within a class, such as the arithmetic mean [34] or mode [35]. Once the prototypes are established, both the prototypical objects and the objects lying in closest resemblance to the prototype (or proximity, in the case of a feature space) can be recognised almost instantly by humans. This phenomenon is also known, in the visual cognition field, as the prototype effect.

Deep image matching has been applied successfully to object recognition tasks [12,36], including robotic applications [13]. In [12] pairs of images belonging to the same dataset, or visual domain, were binary-classified (i.e., as similar/dissimilar) based on the sigmoid activation of the L1 component-wise distance between the two image embeddings representing the pair. These embeddings are produced by two Convolutional Neural Network (CNN) branches, which share the same weights. This characteristic of the proposed architecture, also known as Siamese Network, ensures that: (i) the Network is symmetric with respect to the order chosen to feed the input image pairs, and (ii) predictions are consistent for significantly similar images, since the twin branches are each computing the same function. Furthermore, Zeng et al. [13] have shown that configuring the two CNN branches to learn weights independently can lead to higher performance than that achieved through Siamese-like architectures when images are matched across two different domains, i.e., robot-collected, real-world images and Amazon products. In the following, we refer to the latter configuration as two-branch Network, to differentiate it from Siamese Networks. The study conducted in [13] addressed the second challenge (i.e., the paucity of training examples), by using a pre-trained ResNet [37] for all the Image Matching methods under comparison. In a prior-based fashion, weights were transferred from the less challenging ImageNet dataset [14], which provides abundant training examples [32].

The object representations achieved in [13], however, were obtained in the context of the 2017 Amazon Challenge and were thus tailored on (i) a domain-specific product catalogue (i.e., Amazon's products), (ii) a specific end-task (i.e. object stowing and grasping).

The recent availability of pre-segmented and annotated large-scale image collections (e.g., [14,16,38,39]) has provided new benchmarks to evaluate object categorisation methods. These data sets are task-agnostic in nature and comprise of common, everyday objects. Compared to everyday object collections such as MS-COCO [39] or NYU-Depth V2 [38], the ShapeNet 3D model base [16] was organised with respect to the WordNet taxonomy [17], thus facilitating the further integration of other open, large-scale knowledge bases adhering to the same taxonomy, like ConceptNet [6], DBpedia [7], Visual Genome [18] and others. Already used for learning object intrinsics [40], and 3D scene understanding [41], ShapeNet has never been used, to our knowledge, for training multi-branch Convolutional architectures on the task of few-shot object recognition. We made a first attempt at capitalising on the visual knowledge provided with ShapeNet as part of our prior work [19], where we explored different colour-based and shape-based shallow representations to match ShapeNet

models by similarity. In what follows, we further extend this investigation to assess whether the Deep representations learned by similarity matching on ShapeNet can: (i) outperform the previously-explored shallow representations, as well as (ii) generalise to new object classes.

All the reviewed image matching methods, on the other hand, require that a support set of examples is provided, at inference time, for each new query image. Motivated by this limitation, Qi et al. [15] have proposed a method to use each newly-incoming query image directly as classification weights on a single-branch CNN. This weight initialisation strategy was named weight imprinting by analogy with how cognitive imprinting develops for humans, who can recognise new objects from the very first exposure, drawing connections from their prior knowledge. Similarly, the image embeddings produced by a CNN, if opportunely scaled and L2 normalised, can be directly added the weight matrix of the classification layer, as soon as the first image of a novel object type is observed. Crucially, they found that the top performance could be achieved when weight imprinting was combined with a further fine-tuning of the CNN. Moreover, averaging the embeddings by class before applying them as weights was proven beneficial, in cases when more than one exemplar per class is available. In our view, this result shows an interesting fit within the prototype theory of cognition [33] and, more specifically, with existence of a correlation between central tendency measures and prototype formation [34,35].

Inspired by these results, we aim at investigating whether embedding normalisation, which weight imprinting can be seen as a special case of, can aid the formation of more effective "prototypes" in the context of Deep metric learning. Weight imprinting was initially designed as an alternative to multi-branch metric learning and, therefore, has never been integrated within two-branch Networks. We interrogate on whether or not these within-class averaged representations can improve fine-tuning also in the case of the two-branch architectures of [13], combining the benefits of both approaches. In the context of binary (or binomial) classification, where weight imprinting cannot be applied by design, we rely on the more general L2 normalisation of the input embeddings.

## 3. Proposed Approach and Contributions

To address the research questions posed in Section 2, we propose the following experimental setup. First, we extend the ShapeNet-based dataset presented in [19] to include additional object models and classes. Moreover, we compare the most noticeable results obtained with the shallow methods explored in [19] with the case of directly reusing deep features transferred from ImageNet, without re-training.

Furthermore, we develop a two-branch architecture which combines the ablation, among the ones in [13], including a softmax classification component with the weight imprinting strategy of [15]. To compare with the other Siamese-based ablations of [13], where weight imprinting is not applicable by definition, we instead introduce an L2 normalisation operation before the image embeddings are further compared.

To the best of our knowledge, this experimental setup involves a novel combination of methods and integrates knowledge from a rather unexplored, task-agnostic dataset.

## 4. Data Preparation

ShapeNet (https://www.shapenet.org/taxonomy-viewer) is a large-scale collection of richly annotated 3D models [16] split into: (i) ShapeNetCore, with 55 object classes with about 51,300 unique 3D models, and (ii) ShapeNetSem, consisting of 12,000 more densely-annotated models across 270 categories. For most 3D models, 2D views of the object surfaces are also available. Categories of objects in ShapeNet reference object classes in the WordNet lexical database [17]. We capitalised on the object models available in ShapeNet to derive three datasets, as follows.

**ShapeNetSet1 (SNS1).** The first set, comprising of ten object classes and two models for each class, was compared against the NYUDepth reference dataset [38] as part of the seminal trials in [19].

**ShapeNetSet2 (SNS2).** The second dataset extends SNS1 and provides ten 2D views for each of the ten object classes, allowing us to test performance on different instances of a class (e.g., different models of a sofa), and control for class imbalance, as highlighted by the cardinalities in Table 1.

**ShapeNet+Google-20class (SNG-20).** The third set includes twenty object classes and twenty views per class, split so that, for each class, ten views are devoted to training, five to validation and five to test. It includes objects which can be typically found in office environments—e.g., chairs, piles of paper, computer monitors, as well as object categories that are more specific to the health and safety domain—e.g., fire extinguishers, fire exit signs, power outlets. Whenever the 2D views available in ShapeNet were not sufficient, these were complemented with models retrieved from Google Images (https://github.com/hardikvasa/google-images-download), following "class name no background" as general protocol for search keywords (e.g., office chair no background or potted plant no background). This precaution allowed us to select white-background images, comparable to the models available in ShapeNet. To test the generalisability to new objects, we treat ten object classes as known and the remainder ten as novel. All the tested models are based on ResNet50 CNN branches, pre-trained on the ImageNet-1000 subset. Thus, we formed the two object groups by alternating categories that were already part of ImageNet-1000, with completely novel classes (Table 2).

**Table 1.** 10-Class Datasets Statistics.

| Object | SNS1 | SNS2 |
|---|---|---|
| Chair | 14 | 10 |
| Bottle | 12 | 10 |
| Paper | 8 | 10 |
| Book | 8 | 10 |
| Table | 8 | 10 |
| Box | 8 | 10 |
| Window | 6 | 10 |
| Door | 4 | 10 |
| Sofa | 8 | 10 |
| Lamp | 6 | 10 |
| Total | 82 | 100 |

**Table 2.** 20-Class Datasets Statistics. Novel classes with respect to the ImageNet-1000 dataset are highlighted.

| Object | | SNG-20 Train | SNG-20 Val | SNG-20 Test |
|---|---|---|---|---|
| Known | Chair | 10 | 5 | 5 |
| | Bottle | 10 | 5 | 5 |
| | **Paper** | 10 | 5 | 5 |
| | Book | 10 | 5 | 5 |
| | Desk | 10 | 5 | 5 |
| | **Box** | 10 | 5 | 5 |
| | Window | 10 | 5 | 5 |
| | **Emergency exit sign** | 10 | 5 | 5 |
| | **Coat rack** | 10 | 5 | 5 |
| | Radiator | 10 | 5 | 5 |
| Novel | **Fire extinguisher** | 10 | 5 | 5 |
| | Desktop PC | 10 | 5 | 5 |
| | Electric heater | 10 | 5 | 5 |
| | Lamp | 10 | 5 | 5 |
| | **Power cable** | 10 | 5 | 5 |
| | Monitor | 10 | 5 | 5 |
| | **Person** | 10 | 5 | 5 |
| | **Plant** | 10 | 5 | 5 |
| | Bin | 10 | 5 | 5 |
| | Door | 10 | 5 | 5 |
| | **Total** | **200** | **100** | **100** |

## 5. Few-Shot Image Matching

### *5.1. Model Architecture*

In the tested architecture (Figure 1), pairs of similar and dissimilar images are fed to two twin networks sharing the same weights. Indeed, we hypothesise that tying the weights of the two branches can better fit the case of matching images belonging to the same domain, i.e., ShapeNet in this case. Compared to the traditional Siamese architecture originally proposed in [12], this model introduces some key differences. First, to compensate for the likelihood of metric collapse, we include a ResNet50 pre-trained on ImageNet in each branch, following the approach of [13]. Second, we apply L2 normalisation on the output embeddings returned by each branch and compute their L1 component-wise distance before applying any further transformation. Let $E_i = [e_1, ..., e_k] \in \mathbb{R}^k$ be the k-dimensional embedding resulting from the i-th branch, where $i = 1, .., N$ (i.e., in this configuration $N = 2$ and $k = 2048$). Then the L2-normalised embedding $\hat{E}_i$ can be defined as:

$$\hat{E}_i = \frac{E_i}{|E_i|}. \tag{1}$$

The denominator in Equation (1) indicates the L2-norm of vector $E_i$:

$$|E_i| = \sqrt{\sum_{j=1}^{k} |e_j|^2}. \tag{2}$$

The L1 (or Manhattan) distance between the L2-normalised vectors $\hat{E}_1 = [e_{1,1}, e_{2,1}, ..., e_{k,1}]$ and $\hat{E}_2 = [e_{1,2}, e_{2,2}, ..., e_{k,2}]$ coming from the two branches under consideration is thus the vector $D = [d_1, d_2, ..., d_k]$, where:

$$d_j = |e_{j,1} - e_{j,2}|. \tag{3}$$

In the tested configuration, the L1 distance layer is followed by two fully-connected layers with a ReLU activation function applied in between. A fully-connected layer is a function $f : \mathbb{R}^k \rightarrow \mathbb{R}^M$, which transforms an input vector $D = [d_1, d_2, ..., d_k]$ so that $f(z) = f(b + \sum_{j=1}^{k} d_j w_j)$, where $W = [w_1, w_2, .., w_k]$ and $b$ are the weight vector and bias parameters of the layer. In the case of ReLU, the input $D$ is transformed as :

$$f = max(0, z_m), \tag{4}$$

for $m = 1, ..., M$. In this configuration, we set $M = 2$ for the second fully-connected layer, i.e., framing the classification problem as a binary choice between two classes (similar/dissimilar). Another difference introduced, compared to the original Siamese of [12] was applying a softmax layer to the resulting vectors, instead of a sigmoid one. By definition, a softmax function $\sigma : \mathbb{R}^M \rightarrow \mathbb{R}^M$ is used to map the non-normalised output of a prior layer ($z$) to a probability distribution over $M$ classes, as follows:

$$s_m = \sigma(z)_m = \frac{e^{z_m}}{\sum_{m=1}^{M} e^{z_m}}. \tag{5}$$

The chosen training routine is based on the cumulative binary crossentropy loss across positive and negative mini-batches. In other words, before each parameter update, the model is fed with a mini-batch of positive examples (i.e., similar image pairs), followed by a mini-batch of negative examples (i.e., dissimilar image pairs), to balance out the contribution of each set to the overall loss. Then, the cumulative loss ($L$) for positive and negative outcomes is given by:

$$L = L_+ + L_-, \tag{6}$$

where $L_+$ is the binary cross-entropy loss for the positive sample and $L_-$ is the same metric computed on the negative sample. For a single example, the loss $l$ is defined as:

$$l = -\sum_{m=1}^{M} t_m \log s_m, \tag{7}$$

where $t_m$ is the ground truth value for class $m$. In the case of binary classification, $M = 1, 2$, $t_2 = 1 - t_1$ and $s_2 = 1 - s_1$. Then, the mini-batch loss ($L_+$ or $L_-$ respectively) is obtained by averaging the $l$ values across observations. In Figure 1, for instance, a chair is compared with another model of chair. The loss obtained in this pass would then be summed up to the loss obtained when comparing chairs against desks, for example.

At test time, the same approach as [13] is followed: objects are classified based on their L2 Nearest Neighbour (NN) in the learned feature space, i.e., based on the class label of the nearest embedding, in terms of L2 (or Euclidean) distance. In our example (Figure 1), when a chair is spotted at test time, it is mapped closely to the other chair models observed during training. Moreover, when a novel object instance is observed (e.g., a fire extinguisher) other support examples depicting the same class are provided to update the feature space.

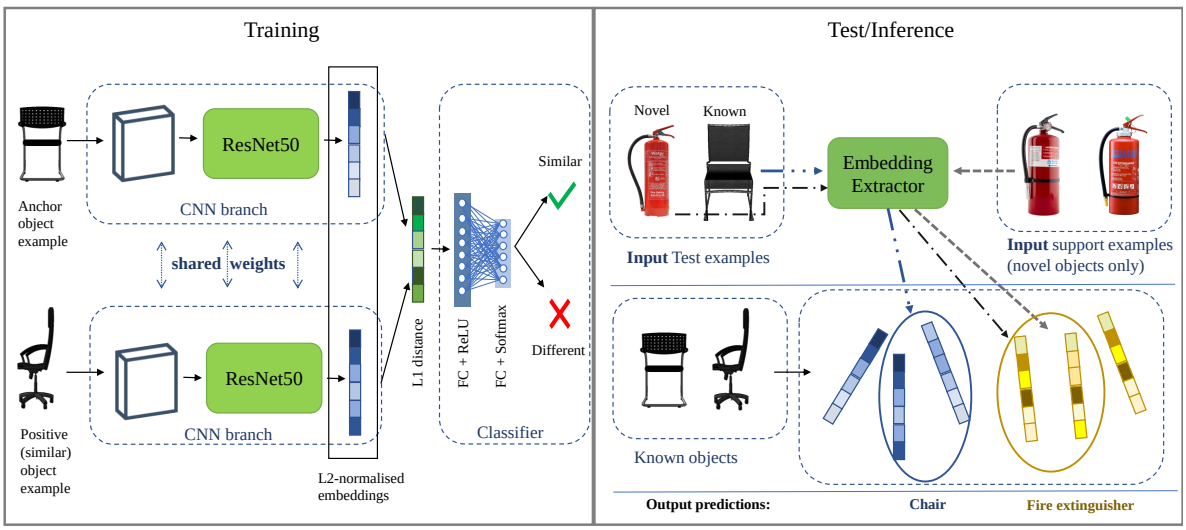


**Figure 1.** Object Recognition workflow. A Siamese Network comprising two ResNet50 CNN branches is trained on known object classes. The resulting 2048-dimensional feature embeddings are first L2-normalised and then combined through their component-wise L1 distance. The two fully-connected layers operate as a binary classifier to conclude on the similarity of each image pair. At test time, objects are classified based on the L2 nearest neighbour in the obtained feature space, comprising of: (i) the embeddings learned during training and (ii) the embeddings extracted for the input novel support examples.

### *5.2. Implementation Details*

The input images were resized as 224 × 224 frames and normalised according to the ImageNet-1000 dataset distribution. We fine-tuned the ResNet50 CNN branches with a mini-batch Gradient Descend optimiser over batches of 16 image pairs, with learning rate set to 0.0001, momentum 0.9 and weight decay 0.00001. The parameter update was repeated for a maximum of 1000 epochs, with an early stopping condition for all cases where the loss on the validation set had not decreased for more than 20 successive epochs.

Our code was developed in the Python3.6 programming language. Specifically, we relied on the Pytorch framework to re-implement the Deep Neural architectures presented in [13]. All experiments

were run on a Dell PowerEdge R740xd server, with an Intel Xeon Silver processor running at 2.20 GHz (at test time) and a Tesla P40 GPU (at training time). We extended the Deepo Docker image (https://hub.docker.com/r/ufoym/deepo/), to simulate a Ubuntu 18.04 OS, CUDAv.9.0 and CUDNNv.7 environment and handle all the other library dependencies (more details can be found in the dockerfile available at our Github repository).

### *5.3. Model Evaluation*

We evaluate the explored architectures on both known and novel object classes, after fine-tuning on the ten known object classes of the SNG-20 train dataset split. At inference time, the learned embeddings are combined with support sets for the ten novel classes in the SNG-20 training split. The following ablation study details the different configurations under comparison and the architectural components modified before each performance assessment.

**Baseline NN** is the chosen baseline, where features are extracted from a ResNet50 pre-trained on ImageNet without further fine-tuning, and, for each extracted image embedding in the test set, labels are predicted through finding the Nearest Neighbour based on the L2 distance.

**N-net** is our implementation of the architecture leading to the best performance on novel object classes, among the ones presented in [13]. N-net is a two-branch Network, i.e., weights are learned independently on each branch, differently from Siamese-based models, where weights are tied across the two twin Networks (see also Section 2). Each CNN branch is a ResNet50 pre-trained on ImageNet. At training time, the Network is fed with data triplets formed so that each training example (anchor) is compared against a positive (similar image) and a negative example (dissimilar image). Here, for each input image, the anchor embedding to compare against is automatically-picked, based on its L2-distance from the input embedding (a technique referred to as "multi-anchor switch" in the original paper [13]). Then, parameter update is based on the objective of minimising the Triplet Loss [42]. In this way, the Network optimisation is directed towards minimising the L2-distance between matching pairs, while also maximising the L2-distance between dissimilar pairs.

**K-net** was derived from N-net, by summing up a second component based on softmax classification (see also Equations (5) and 7)) to the Triplet Loss. Zeng et al. [13] found that this choice helped to boost performance on known object classes, at the expense of novel object recognition. Our implementation of both the K-net and the N-net architectures only differs by the fact that we have fine-tuned both ResNet50 branches. In the original design, the Network is fed with images belonging to two different visual domains, i.e., the robot-collected images and the product catalogue images. Then, only the CNN branch devoted to the robot-collected frames is re-trained, while in the product image branch features are extracted without further re-training. In our case, however, where both data streams consist of synthetic object models, if we simply extracted features from both branches without re-training, the resulting architecture would have reduced to the already tested baseline NN.

**SiamResNet50** follows the same general architecture of [12], but relies on a pre-trained ResNet50 for each CNN branch. The main differences with K-net and N-net [13] are that, in this case, (i) weights were shared across the two CNN streams and (ii) training is based on a binary classifier, regularised through a cross-entropy objective (i.e., the same training routine explained in Section 5.1).

**Imprinted K-net** is a K-net where the classification layer used as auxiliary component for the training loss is initialised with imprinted weights, instead of random weights. The weight imprinting approach proposed in [15] is based on the intuition that the embedding layer activations of a CNN, if opportunely scaled and L2-normalised, are mathematically comparable to the weight vectors of a softmax classification layer and can thus be added directly to the classifier's weight matrix, as soon as a new labelled (training) example is collected. We refer to [15] for a complete mathematical demonstration of this argument. Following the same notation introduced in Section 5.1, let $E_i$ be the vectorised representation of an input training point, $m \in M$ its related class label, $W = [w_1, w_2, .., w_k]$

and $b$ the weight vector and bias parameters of the softmax activation layer used to classify the input over $M$ classes. In the weight imprinting routine, each $E_i$ is L2-normalised, yielding $\hat{E}_i$ (see also Equation (1)) and then multiplied by a scaling factor $sf$. Furthermore, $b$ is set to zero in the softmax classification layer. Then, the $m$-th column of the $W$ matrix ($w_m$) is initialised with $\hat{E}_i$. If $n > 1$ training examples are available for class $m$, i.e., as in the case of this experiment, first the average embedding of all $\hat{E}_i$ in class $m$ is computed as

$$E_{i,avg} = \frac{1}{n} \sum_{i=1}^{n} sf\hat{E_i^{(i)}}; \tag{8}$$

then $E_{i,avg}$ is L2-normalised again to $\hat{E_{i,avg}}$ . Finally, $w_m$ is initialised to $\hat{E_{i,avg}}$.

**L2normL1 + SiamResNet50** is the architecture in Figure 1. It is based on SiamResNet50, with the exception that the embeddings output of each branch is L2-normalised before any further transformation. A more detailed description of this ablation is provided in Section 5.1.

## 6. Results and Discussion

The results obtained in [19], which are also summarised in Table 3, albeit providing an improvement from random label assignment in all configurations, were not satisfactory to discriminate different object classes. After transferring weights from a ResNet architecture, pre-trained on abundant natural images depicting different objects [14], we obtained a significant improvement in the object recognition accuracy (as shown in Table 3). Indeed, introducing Deep CNNs and transfer learning helped to separate the feature space. Given the promising results obtained even with the baseline solution, we included weights learned on ImageNet also in all the other tested models.

**Table 3.** Cumulative accuracy results, when views in SNS1 are matched against SNS2.

| Approach | Dataset SNS1 v. SNS2 |
|---|---|
| Random label assigment | 0.10 |
| Shape-only L3 | 0.19 |
| Colour-only Hellinger | 0.32 |
| Shape+Color (weighted sum) | 0.32 |
| Baseline NN | **0.82** |

The test results of all experimented solutions on both known and novel object classes are summarised in Table 4. The baseline NN feature space was already 80% accurate in the mixed scenario but degraded to a sub-optimum when fine-tuning the alternative solutions derived from [12] and [13]. Thus, we can speculate that, with the few examples available, it was challenging to separate the starting embedding space further, likely because it included semantically-related class pairs (e.g., electric heaters and radiators, books and paper documents). It is also worth noting that the F1 score obtained when applying the baseline to the unseen classes is higher than in the case of known objects. This evidence may be related to how the object classes assigned to the novel objects were clustered within the initial feature space transferred from ImageNet. One precaution we followed during the data preparation phase was splitting the classes shared in ImageNet equally across the known and novel sets (see Section 4). Nonetheless, the feature space transferred from ImageNet carried over a set of biases which, on average, made the novel classes inherently easier to separate than the known classes. For instance, *books* and *papers* both belong to the known object group but represent highly-related concepts, both semantically and visually. The same behaviour can be observed when looking at the F1 scores of both Siamese-based methods. Even further, in the case of SiamResNet50, the results obtained on known and novel objects were equal to the baseline NN, suggesting that the learned feature space has not diverged much from the initial one. The fact that L2norm L1 SiamResNet50 also performed better on novel objects than on seen categories can be explained by

considering that both ablations share the same backbone layers and loss function. The introduction of the embedding L2-normalisation is, in fact, their most differentiating trait.

**Table 4.** Test results obtained on the SNG-20 set, when evaluating on top-1 matches. Measured in terms of Accuracy, Precision, Recall and F1 score. Each test example is provided with a 10-image support set, at inference time.

| Approach | Known | | | | Novel | | | | Mixed | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **A** | **P** | **R** | **F1** | **A** | **P** | **R** | **F1** | **A** | **P** | **R** | **F1** |
| Baseline NN | 0.76 | 0.91 | 0.76 | 0.79 | **0.84** | 0.93 | **0.84** | 0.85 | 0.80 | 0.85 | 0.80 | 0.79 |
| N-net [13] | 0.70 | 0.97 | 0.70 | 0.78 | 0.74 | 0.83 | 0.74 | 0.75 | 0.72 | 0.79 | 0.72 | 0.72 |
| K-net [13] | 0.76 | 0.94 | 0.76 | 0.81 | 0.70 | 0.78 | 0.70 | 0.73 | 0.78 | 0.73 | 0.73 | 0.73 |
| SiamResNet50 [12,37] | 0.76 | 0.91 | 0.76 | 0.79 | **0.84** | 0.93 | **0.84** | 0.85 | 0.78 | 0.83 | 0.78 | 0.77 |
| Imprinted K-net | 0.74 | **0.98** | 0.74 | 0.82 | 0.80 | 0.86 | 0.80 | 0.82 | 0.77 | 0.83 | 0.77 | 0.77 |
| L2norm L1 SiamResNet50 | **0.80** | 0.91 | **0.80** | **0.83** | **0.84** | **0.95** | **0.84** | **0.87** | **0.82** | **0.86** | **0.82** | **0.81** |

There were no significant differences, in terms of training times, between all methods that involved re-training (i.e., excluding the baseline NN). On the largest data set (SNG-20) and with the hardware configuration described in Section 5.2, all the experimented methods converged in less than 1.5 h of training. In terms of computational complexity, the Imprinted K-net required an additional iteration over the data to compute the average class embeddings. However, we verified that this process could be run efficiently on the GPU, in 0.258 s for the largest data set.

Overall, the Siamese-like architecture derived from [12] led to higher performance than the N-net and K-net alternatives. This result seems to verify our initial hypothesis, i.e., that weight sharing is more suitable for treating images belonging to the same visual domain, and further supports the findings in [13]. The improvement may have also been caused by the different distance function imposed in the two cases. Introducing weight imprinting in the K-net architecture ensured that we could reach near-perfect precision on the known object classes, at the expenses of performance on the novel objects. Ultimately, applying L2-normalisation to the embeddings of a Siamese ResNet50 derived from [12], before the L1 component-wise distance layer, led to the most robust performance, on both known and novel objects.

## 7. Conclusions

In this work, we evaluated different state-of-the-art few-shot image matching methods on a novel data set, based on openly-available data from ShapeNet [16] and Google Images. We found that relying on a pre-trained CNN could provide an improved starting performance, compared to all the shallow approaches explored in [19]. We also tested the effects of (i) imprinting the weights of a two-branch CNN, and (ii) L2-normalising the embeddings of a Siamese CNN. The latter configuration led to the top performance on both known and novel objects. These results appear to be related to the fact that images were matched within the same visual domain. Future work includes testing these methods in a more challenging case, where real-world images collected by a robot in the wild are matched against the Web-retrieved models used in this work.

Moreover, the supporting knowledge used in this work was limited to the visual knowledge transferred from ImageNet, ShapeNet 2D and Google Images, and to the WordNet taxonomy linking these sources. Therefore, we also intend to further explore whether and how the use of different knowledge modalities can support continual object learning in the case of mobile robots.

**Author Contributions:** Conceptualization, A.C., G.B., E.B. and I.T.; Data curation, A.C. and G.B.; Formal analysis, A.C.; Funding acquisition, E.B., I.T. and E.M.; Investigation, A.C.; Methodology, E.B.; Software, A.C. and G.B.; Supervision, P.M. and E.M.; Visualization, A.C.; Writing—original draft, A.C.; Writing—review & editing, I.T. and E.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has been partially supported by a European Union's Horizon 2020 grant—Sciroc, No 780086.

**Acknowledgments:** The authors would like to thank Prof. Stefan Rueger (Knowledge Media Instutite, The Open University, UK), for graciously providing the computing resources used in this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. SPARC. Strategic Research Agenda for Robotics in Europe 2014–2020. EU Robotics, 2014. Available online: https://www.eu-robotics.net/cms/upload/topic_groups/SRA2020_SPARC.pdf (accessed on 21 February 2020).
2. Tiddi, I.; Bastianelli, E.; Daga, E.; d'Aquin, M.; Motta, E. Robot–City Interaction: Mapping the Research Landscape—A Survey of the Interactions Between Robots and Modern Cities. *Int. J. Soc. Robot.* **2019**, 1–26. [CrossRef]
3. Bastianelli, E.; Bardaro, G.; Tiddi, I.; Motta, E. Meet HanS, the Health&Safety Autonomous Inspector. In Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks, 17th International Semantic Web Conference (ISWC 2018), CEUR Workshop Proceedings, Monterey, CA, USA, 8–12 October 2018.
4. Mollaret, C.; Mekonnen, A.A.; Pinquier, J.; Lerasle, F.; Ferrané, I. A multi-modal perception based architecture for a non-intrusive domestic assistant robot. In Proceedings of the The 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Christchurch, New Zealand, 7–10 March 2016; pp. 481–482.
5. Ferri, G.; Manzi, A.; Salvini, P.; Mazzolai, B.; Laschi, C.; Dario, P. DustCart, an autonomous robot for door-to-door garbage collection: From DustBot project to the experimentation in the small town of Peccioli. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 655–660.
6. Speer, R.; Chin, J.; Havasi, C. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4444–4451.
7. Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. Dbpedia: A nucleus for a web of open data. In *The Semantic Web*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 722–735.
8. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
9. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence;* IEEE Computer Society: Washington, DC, USA, 2017; Volume 39, pp. 1137–1149; doi: 10.1109/TPAMI.2016.2577031. [CrossRef]
10. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
12. Koch, G.; Zemel, R.; Salakhutdinov, R. Siamese neural networks for one-shot image recognition. In Proceedings of the ICML Deep Learning Workshop, Lille, France, 10–11 July 2015; Volume 2.
13. Zeng, A.; Song, S.; Yu, K.T.; Donlon, E.; Hogan, F.R.; Bauza, M.; Ma, D.; Taylor, O.; Liu, M.; Romo, E.; et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 1–8.
14. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2009; pp. 248–255.
15. Qi, H.; Brown, M.; Lowe, D.G. Low-shot learning with imprinted weights. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 5822–5830.
16. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012.
17. Miller, G.A. WordNet: A lexical database for English. *Commun. ACM* **1a995**, *38*, 39–41.

18. Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.J.; Shamma, D.A.; et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.* **2017**, *123*, 32–73.
19. Chiatti, A.; Bardaro, G.; Bastianelli, E.; Tiddi, I.; Mitra, P.; Motta, E. Exploring Task-agnostic, ShapeNet-based Object Recognition for Mobile Robots. In Proceedings of the EDBT/ICDT 2019 Joint Conference, Lisbon, Portugal, 26 March 2019.
20. Chen, Z.; Liu, B. Lifelong machine learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2018**, *12*, 1–207.
21. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**. [CrossRef]
22. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
23. French, R.M. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **1999**, *3*, 128–135. [PubMed]
24. Grossberg, S.T. *Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 70.
25. Shin, H.; Lee, J.K.; Kim, J.; Kim, J. Continual learning with deep generative replay. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 2990–2999.
26. Aljundi, R.; Belilovsky, E.; Tuytelaars, T.; Charlin, L.; Caccia, M.; Lin, M.; Page-Caccia, L. Online continual learning with maximal interfered retrieval. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; 2019; pp. 11849–11860.
27. Tenorth, M.; Beetz, M. Representations for robot knowledge in the KnowRob framework. *Artif. Intell.* **2017**, *247*, 151–169.
28. Nolfi, S.; Parisi, D. Learning to adapt to changing environments in evolving neural networks. *Adapt. Behav.* **1996**, *5*, 75–98.
29. Rusu, A.A.; Rabinowitz, N.C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. Progressive neural networks. *arXiv* **2016**, arXiv:1606.04671.
30. Hinton, G.E.; Salakhutdinov, R.R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504–507. [PubMed]
31. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
32. Huh, M.; Agrawal, P.; Efros, A.A. What makes ImageNet good for transfer learning? *arXiv* **2016**, arXiv:1608.08614.
33. Rosch, E.; Mervis, C.B.; Gray, W.D.; Johnson, D.M.; Boyes-Braem, P. Basic objects in natural categories. *Cogn. Psychol.* **1976**, *8*, 382–439.
34. Posner, M.I. Abstraction and the process of recognition. *Psychol. Learn. Motiv.* **1969**, *3*, 43–100.
35. Neumann, P.G. Visual prototype formation with discontinuous representation of dimensions of variability. *Mem. Cogn.* **1977**, *5*, 187–197.
36. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
38. Nathan Silberman, Derek Hoiem, P.K.; Fergus, R. Indoor Segmentation and Support Inference from RGBD Images. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2012.
39. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
40. Shi, J.; Dong, Y.; Su, H.; Stella, X.Y. Learning non-lambertian object intrinsics across shapenet categories. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5844–5853.

41. Xiang, Y.; Fox, D. DA-RNN: Semantic mapping with data associated recurrent neural networks. *arXiv* **2017**, arXiv:1703.03098.
42. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.